

(19)



Europäisches Patentamt
European Patent Office
Office européen des brevets



(11)

EP 0 797 881 B1

(12)

EUROPEAN PATENT SPECIFICATION

(45) Date of publication and mention
of the grant of the patent:
30.08.2000 Bulletin 2000/35

(51) Int Cl.⁷: **H04L 12/413, B60R 16/02,
H04L 29/10, H04L 25/49**

(21) Application number: **95943428.3**

(86) International application number:
PCT/US95/16246

(22) Date of filing: **15.12.1995**

(87) International publication number:
WO 96/19058 (20.06.1996 Gazette 1996/28)

(54) **COMMUNICATIONS NETWORK, A DUAL MODE DATA TRANSFER SYSTEM THEREFOR**
KOMMUNIKATIONSNETZ MIT DOPPELMODUS-DATENTRANSFERSYSTEM
RESEAU DE TELECOMMUNICATIONS ET SON SYSTEME DE TRANSFERT BIMODAL DE
DONNEES

(84) Designated Contracting States:
AT DE GB

(30) Priority: **16.12.1994 US 357002**

(43) Date of publication of application:
01.10.1997 Bulletin 1997/40

(73) Proprietor: **DaimlerChrysler Corporation**
Highland Park, Michigan 48288-1919 (US)

(72) Inventors:
• **HALTER, Richard, A.**
Warren, MI 48092 (US)
• **RANDEL, Robert, J.**
Walled Lake, MI 48390 (US)

• **HORMEL, Ronald, F.**
Clinton Township, MI 48038 (US)
• **STARK, John**
Ann Arbor, MI 48105 (US)

(74) Representative: **Baillie, Iain Cameron**
Ladas & Parry,
Dachauerstrasse 37
80335 München (DE)

(56) References cited:
EP-A- 0 246 666 US-A- 5 274 636

• **AUTOMOTIVE ENGINEERING, vol. 100, no. 9, 1**
September 1992 pages 37-40, XP 000310446
'CLASS INTRODUCTION TO MEDIUM-SPEED
MULTIPLEXING'

Note: Within nine months from the publication of the mention of the grant of the European patent, any person may give notice to the European Patent Office of opposition to the European patent granted. Notice of opposition shall be filed in a written reasoned statement. It shall not be deemed to have been filed until the opposition fee has been paid. (Art. 99(1) European Patent Convention).

EP 0 797 881 B1

Description

[0001] This invention relates to communication over a multiplex data communications network. In particular, this invention relates to a system for interchanging data bytes within a node between a microcontroller (MCU) and a first peripheral at a chosen clock frequency. Then within the same node, the first peripheral generates a protocol bit and then converts both the protocol bit and each bit of the data bytes from the MCU into variable pulse width modulated (VPWM) bits forming a VPWM message. After forming the message of VPWM bits, the first peripheral sends the VPWM bits to a second peripheral that converts each VPWM bit into a communications symbol (known as a J1850 symbol) conforming to the communication requirements of the Society of Automotive Engineers (SAE) established Recommended Practice J1850 (a set of technical requirements and parameters).

[0002] The second peripheral, a transceiver, drives these symbols onto a single-wire communications bus to communicate the message to other nodes connected to the bus. The second peripheral also routes a reflection of the symbols back to the first peripheral and reconverts the symbols into VPWM bits. The first peripheral then reassembling the reflected bits into bytes used in the next interchange of bytes with the MCU and to check for the achievement of arbitration if other nodes on the bus are transmitting.

[0003] Normally, transmitting and receiving serial bytes between the first peripheral device and the MCU require interrupts in the transfer programming. U.S. Patent No. 5,274,636 dated Dec. 28, 1993 of Halter et al. entitled "Automatic Multiplex Data Link System, Symbol Encoder Decoder Therefor" (or SED patent) teaches such a system. Such interruptions imposes a heavy burden on the MCU to perform the process of transmitting and receiving serially arranged byte messages. During transmission, programming of the MCU must allow for, illustratively, 34 microseconds between bytes for the introduction of an interrupt. This imposes time restraints on the MCU when preparing each byte for transmission. Similar time restraints occur for receiving bytes of data. We searched for ways to reduce the imposed time restraints on the MCU while improving the processing performance. That search ended in the present invention which accomplishes reducing the time restraints required to process the data prior to converting the data into J1850 symbols and for determining whether a node has gained control of the bus by winning an arbitration contest with competing nodes during a message frame.

[0004] The present invention discloses using a conventional serial peripheral interface (SPI) transfer circuit within the MCU to communicate with a serial peripheral data register (LSPDR) in a peripheral referred to as a link device (Link). The transfer of data bytes between the MCU and the Link occurs in the form of individual bits of a byte transferred serially from a data register in

the MCU to a data register in the link during a SPI transfer (SPI-XFR) at a chosen clock rate illustratively, one bit/one μ s pulse. Then, from the Link, the individual bits of the SPI-XFRed byte are converted to variable pulse width modulated bits and are then driven onto the bus by an integrated driver/receiver (IDR) module as J1850 symbols at the chosen clock rate but delayed to compensate for the variations in pulse width of each symbol, e.g., one symbol may be longer than another (64 or 128 μ s long). Hence 64 clock pulses would occur during the transfer of a 64 μ s long symbol. The ratio of SPI transfer to bus transfer are such that a period of time, illustratively 500 μ s, may result between SPI-XFRs of individual bytes, removing the need for interrupts. The system, however, contains the capability of operating in an interrupt mode.

[0005] The sending and receiving of bits of the data bytes between the Link and the second peripheral called an integrated driver receiver circuit (IDR) and over the single-wire bus occur at the same rates of speed that symbols travel over the bus.

[0006] Clock signal (SPISCK) clocks bits of data bytes through circuits within the MCU at a fixed clock frequency while shift clock (SHCLK), clock (CLK) and clock divide by 2 (CLKDIV2) and clock circuits within the Link and IDR during transfers of VPWM bits and symbols

[0007] The invention and its modes of operation will be more fully understood from the following detailed description when taken with the appended drawing figures in which:

Fig. 1 illustrates in block diagram form a communications network employing the dual-stage data transfer system of this invention in each node of the network;

Fig. 2 depicts in expanded block diagram form to show the components of the Link in a pair of nodes of Fig. 1 connected to the communication bus;

Figs. 3, 4 and 5 illustrate in partial block and logic diagram form the transmitter/receiver, the transmitter and receiver circuits, respectively, of the symbol encoder/decoder modified to interface and form a part of the Link;

Fig. 6A-6G depict significant timing sequences associated with the operation of the SPI-XFR system; Fig. 7A-7J illustrate timing diagrams associated with transferring a three-byte message from the MCU to the Link and over the single-wire bus to other nodes of the network;

Fig. 8 illustrates a state diagram of the various states required to transfer byte messages from the MCU to the bus and for checking arbitration of symbols on the bus;

Fig. 9A-9C illustrates in partial block and partial logic diagram form the devices in Link used to exchange data bytes with the MCU and to send to and receive from the bus data symbols; and

Fig. 10A and 10B depict the flowchart of the

firmware programs for controlling the operation of the MCU during solicited and unsolicited SPI-XFRs of data bytes.

Fig. 10C represents a detail of said flowchart.

[0008] Fig. 1 depicts a block diagram of a communications network 10 with several nodes 16a-16f coupled between a single-wire bus 12 and a plurality of sensors or application devices 14a-14f. The nodes include MCUs 18a-18f and Links 22a-22f circuits and interfaces for performing SPI-XFRs, and IDRs 20a-20f connected to the Links for converting between digital pulses and J1850 symbols during sending and receiving messages over bus 12. Each Link 22 interconnects between an MCU 18 and an IDR 20.

[0009] Nodes 16a-16f, disposed throughout the vehicle connect to bus 12 via stubs 13a-13f, respectively. Bus 12 supports bi-directional transfers of serial data between nodes. Nodes 16a-16f permit exchange of bit stream information of almost any established protocol.

THE NODES

Sensor to MCU Transfers

[0010] Referring to Fig. 2, there a block diagram depicts additional details of nodes 16a and 16b. Illustratively, in node 16a, sensor 14a detects analog measurands and an analog to digital converter in MCU 18a converts that signal into digital signals usable by MCU 18a. MCU 18a accepts and then transforms the digital signals into a series of message bytes for transfer over bus 12. To effect the transfers, each message byte routes over a parallel bus within MCU 18a and into a conventional MCU serial peripheral data register (MSPDR) 19f for temporary storage.

Loading a Message Byte into the Link by a SPI-XFR From MCU

[0011] Assume node 22a desires to transmit a message over bus 12 to other nodes connected to the bus. When prompted, MCU 18a interchanges the stored message byte by a SPI-XFR, bit-by-bit, serially with a previously transferred data byte stored in a serial peripheral data register (LSPDR) 28a of Link 22a. During the initial SPI-XFR, MCU 18a gets unusable data while Link 22a gets the first message byte associated with the sensed data. The interchanging of bits occur via SPI circuit 19d through master-in slave-out (MISO) and master-out slave-in (MOSI) pins of MCU 18a that connects MSPDR 19f to LSPDR 28a. A SPI clock (SPISCK) signal emanating from Link 22a clocks movement of bits through MSPDR 19f while a Link clock signal (SHCLK) clocks movement of bits through LSPDR 28a. SPISCK starts after MCU 18a strobes a handshaking signal called Byte Ready, (BYT RDY) from a designated output port to holding registers STATEIN 62a, STATEOUT 70a

and a main state machine (MSTATE 48a) of Link 22a. The operation and structural details of MSTATE 48a, STATEIN 62a and STATEOUT 70a appear in co-pending patent application, serial number U.S.S.N. 08/356,999 of Halter et. al. filed December 16, 1994.

[0012] Upon receipt of BYT RDY, Link 22a immediately initiates, illustratively, a high-speed 1 MHz, eight-bit, SPI-XFR. MSPDR unloads byte 1 through the MISO pin transferring serially one bit at a time of the first byte of the message to LSPDR. Simultaneously, LSPDR serially unloads bits of byte X (unknown data) into the MOSI pin of MCU 18a. Figs. 6E and 6F show the SPI-XFR in response to SPISCK of Fig. 6C. The rising edge of SPISCK causes MCU 18 to place data at the sending MISO pin and at the sending MOSI pin of Link 22. The falling edge of SPISCK causes the latching of the data on the receiving pins of MCU 18a and Link 22a. The transfer of each byte consumes about 11 μ s. Following the transfer of byte 1, and the receipt of an unusable byte X, MCU 18a writes the next byte, byte 2, over byte X in MSPDR 19f and then strobes BYT RDY again. The rising edge of this second BYT RDY must occur at least 2.1 μ s after the falling edge of the last SPISCK pulse of the SPI transfer. (See Fig. 7D and 7G Index # 2).

Link and IDR converting the SPI Transferred Data Bits into Data Symbols

[0013] The most significant bit (MSB) and then each succeeding bit of byte 1 in LSPDR 28a routes to an Encoder 37a of Fig. 2. Encoder 37a converts each bit into send codes SND0 and SND1. These send codes have values of 00, 01, 10 or 11 depending upon the bit value and level (HIGH or LOW) of the bus for the next symbol.

[0014] Send codes route to ports of the XMIT/REC Ckt 24 of Fig. 3 of the symbol/encoder (SED) 23a and join (not shown) an additional code (a logic 0 code or a fixed ground signal) to form a three-bit-send code. This three-bit-send code joins with a three-bit-receive code (not shown) to form a six-bit code that routes to a 6 to 3 ROM address multiplexer (MUX) 38. As described in the SED patent, U.S. Patent No. 5,274,636, each data byte bit from the SPI-XFR causes an address from MUX 38 of XMIT/REC Ckt 24 in SED 23 to select one of eight 10 bit transmit words stored in a decoder ROM 40. These 10 bit words represent nominal lengths of time for the various J1850 symbols (minus 19 μ s). It takes about 19 μ s for SED 22a of Fig. 2 to receive back from IDR 20a the reflection of the leading edge of the transmitted symbol. Primarily, the messages will contain symbols for the start-of-frame (SOF) (also referred to as start-of-message (SOM)), a long bit (nominally 128 μ s), a short bit (nominally 64 μ s), an end-of-data (EOD) and an end-of-frame (EOF) symbol which are nominally 200 and 239 μ s respectively.

Converting Nominal Length Words into J1850 Symbols

[0015] Within the XMIT/REC Ckt24, a comparator 44 compares time counted in a 10-bit counter 42 to the value of the ROMDATA word from the 4 to 10 Decoder ROM 40 to determine if a comparison occurs. If so, then circuits in Transmit (XMIT) Ckt 26 of Fig. 4 ends the transmission of that symbol. The link then starts the transmission of the next symbol and initiates a variable pulse width modulated (VPWMout) signal of the selected width and logic level from SED 23a of Fig. 2 to the transmitter (XMIT) circuit of an interface driver receiver (IDR) circuit 20a. The XMIT ckt of IDR 20a converts each VPWMout signal into a trapezoidal pulse shaped symbol resembling an analog signal. These symbols travel over bus 12 at a rate of about 10.4 K bits/sec. or 10.4 Kilobauds.

Checking Arbitration

[0016] A receiver circuit in IDR 20a taps single-wire bus 12 and reflects each symbol back towards transmitting MCU 18a. For data symbols, not protocol symbols, this reflection of symbols occurs during state 10 (ST10), explained infra (or see Fig. 8, of the main state machine in Link 23a). Arbitration of the symbols on bus 12 occurs with respect to a BITVAL signal from SED 23a. (The transfer and arbitration of protocol symbols are explained in co-pending application of Halter et. al. filed December 16, 1994, serial no. U.S.S.N. 08/356,999).

[0017] A change to ST10, the main state for bus 12 transmitting and receiving bits to and from bus 12, causes the XMITMODE of STATEIN 62a to set and to initiate transmission of a symbol bit onto bus 12. The receiver circuit of IDR 20a receives a reflection of the symbol placed on bus 12 and converts the symbol to a digital bit and routes the bit back to a receive pin of link 22a. The receive pin connects to REC Ckt 27 of Fig. 5 within SED 23a. The REC Ckt 27 contains circuits that determine whether a logic 1 or 0 bit value (BITVAL) relates to the reflected symbol. Then STATEIN holding register 62a of Fig. 2 compares the BITVAL received from SED 23a to the previous MSB of LSPDR 28a. (See Fig. 9B). If the same bit comes in that was previously the MSB of LSPDR 28a, then Link 22a wins arbitration and moves to ST 12. If not, Link 22a loses arbitration to another node on bus 12. (See Fig. 7H, I, and J that illustrates which node wins arbitration).

[0018] In ST 12, the arbitrated bit shifts into LSPDR 28a and the BITCNT in STATEIN 62a increments. (See 9C). If the BIT CNT does not equal zero, then MSTATE 48a moves back to ST 10. Arbitration of symbols and the movement between ST10 and ST12 continues until all symbols in the message byte passes. The Link clock signal, SHCLK, clocks the bit movement during ST 10 and ST12. Fig. 6G illustrates the reception of logic 1 and logic 0 bits from bus 12. The time between SHCLK pulses is determined by the duration of the pulse. A logic 0

could be a long or short symbol, hence the duration of the pulse may be either 64 or 128 μ s. From ST 12, if BIT CNT equals zero, then MSTATE 48a moves to ST 13.

5 SPI TRANSFER FOR UNLOADING TO THE MCU THE RECEIVED BYTE AND LOADING THE NEXT BYTE TO THE LINK

[0019] In ST 13, node 16a prepares for a SPI transfer after receiving eight reflections of bits from bus 13 and placing them in LSPDR 28a.

[0020] In ST14, the SPI-XFR between Link 22a and MCU 18a occurs in response to the 1 μ s SPISCK. By using BIT CNT, Link 22a and MCU 18a interchange the eight bits in the LSPDR 28a with the eight bits of byte 2 in MSPDR 19f.

TRANSMITTING THE NEXT BYTE ONTO THE BUS

20 [0021] In ST 15, link 22a clears the SPI-XFR holding register in STATEOUT 70a due to the completion of the SPI-XFR. Assuming BYTE 2 does not represent the LAST BYTE and MCU 18a has indicated another BYT RDY (BYTE 3), MSTATE 48a moves to ST17, the normal transmitting path.

25 [0022] In ST17, the bits of BYTE 2 are transmitted over bus 12. BYT RDY holding register in STATEIN 27a is cleared, and the LAST BYTE shift register is shifted. The shifting of the LAST BYTE is necessary because its status is associated with the data byte that was most recently shifted in from MCU 18a. Upon completion of the message, a cycle redundancy check (CRC) of the transmitted characters is performed (ST15, 19, 20, 21, 10). After the (CRC), an end of frame (EOF) occurs (ST10, 23, 24).

TIMING WITHIN LINK

40 [0023] As noted in Fig. 2, SED 23 provides clock signals for the node. An OSC signal from MCU 18 provides a source of continuous and accurate clock signals used for clocking system operations.

THE CLK SIGNAL.

45 [0024] After dividing OSC into OSCDIV2 and OSCDIV4, a multiplexer (MUX) circuit (not shown) extracts from the two pulse trains a first clock signal (CLK) of Fig. 6A of a chosen frequency, e.g., 2 MHz and the inverse clock signal (CLKL).

THE CLKDIV2 SIGNAL

50 [0025] Within SED 23, a sequential logic divide by two () circuit converts, illustratively, the 2 MHz CLK signal to a 1 MHz CLKDIV2 signal of Fig. 6B that routes to various sequential logic circuits of Link 22. CLKDIV2 signals control the transfer of input signals into Link 22 and the

changing of the states. Note in Fig. 9A, the rising edge of the CLKDIV2 signal clocks memory 33b.

THE INTCLK SIGNAL

[0026] A combinational logic circuit converts CLKDIV2 to a 1 MHz INTCLK signal of FIG. 6C offset from CLKDIV2 by a 1/4 cycle. The INTCLK signal clocks the setting of flags and other output signals from Link 22 including SO-0 through SO-33. (See Fig. 9A). INTCLK combines with other control signals to create additional control signals such as SPICLK that occur at the same time as INTCLK.

THE STENH SIGNAL

[0027] Another delay circuit converts the INTCLK signal into a 1/4 cycle pulse occurring in the last 1/4 of the INTCLK cycle. The STENH signal, Fig. 6D, provides the clock signal for gating the state output signals.

THE OPERATION OF THE SYSTEM

[0028] To describe the operation of the system when transferring data bytes, refer to Fig. 7 through Fig. 10B. In Fig. 9A, a sensor 14 supplies variable measurand information such as pressure, rate of flow, speed etc. in digital form to MCU 18. MCU 18, illustratively, which could be a microcontroller such as the M68HC05 of Motorola or an equivalent, converts the digital bits from sensor 14 into message frames. J1850 messages require a protocol symbol (SOF) to precede a header byte and one or two protocol symbols (EOD and EOF) to follow the cyclic redundancy check (CRC) bytes in a message frame. The manner this MCU/Link system deals with protocol symbols is explained in the copending application mentioned supra.

[0029] With reference to Fig. 7, Fig. 7A shows the sequence of numbers of the state machine flow diagram of Fig. 8 in an effort to coordinate the operation of the system. Fig. 7B shows an idle condition of bus 12 and that bus 12 has been idle for some period of time before MCU 18 initiates a transmission. This means that when bus 12 became idle, MCU 18 did not have a message queued up for transmission.

Solicited Transfer: SPI-XFR Initiated by MCU

[0030] When MCU 18 obtains a message for transmission, MCU 18 attempts to perform a solicited transfer. Such a transfer occurs when MCU 18 transmits an initial message byte, usually the header byte. MCU 18 enters a transmitting mode by not setting the serial peripheral interface status flag (SPIF) in Fig. 10A of the serial peripheral status register (SPSR) or as a result of a reset of the system. MCU 18 asserts the BYT RDY pin to start the transmission of a message.

[0031] After the solicited transfer, SPIF gets set auto-

matically by MCU hardware. Then unsolicited transfers occur for all subsequent transfers between Link 22 and MCU 18.

[0032] Figs. 10A and 10B illustrate in flowchart form, the software operations in MCU 18 used to communicate with link 22. These charts represent an effort to organize the generation and flow of signals through MCU 18. These flowcharts differ from the conventional box and diamond flowcharts. In these charts, signals entering and leaving the flow of control pass along lines and proceed down the page. Conventional start of program and end of program are used. Trapezoidal or rectangularly shaped boxes indicate signals entering and leaving the flow of control. Actions are signified by a small black circle on the lines, the description of the action being given to the left or right of the flow lines. When introducing a decision, the flow line branches to the right or left. A branch may be caused by either a YES or NO condition, with these being signified by a Y or N in a small box as indicated. An arrowhead depicts where secondary flow rejoins the main flow. A reference number indicates the position or step along the line being discussed.

[0033] Returning now to the solicited transfer, note in Fig. 10A, this firmware program requires MCU 18 to determine the status of SPIF. Assuming a set SPIF, MCU 18 then must determine the status of bus 12. Link 22 provides an idle bus status signal to MCU 18. For an idle bus 12, CPU 19a receives instructions to determine the status of the input capture flag (ICF). A set ICF signal means an idle bus has occurred since the last time MCU 18 inspected bus 12, i.e., there is an outstanding unprocessed idle signal. For example, MCU 18 was receiving a message, for which the transmission has finished. MCU 18 must process the received message before starting the transmission, due to time considerations. Assuming no ICF, then CPU 19a determines whether the XMTNUM RAM counter, used to count the number of bytes in the transmit buffer (XMTBUF), has a non-zero value. Using the message depicted in Fig. 7C, illustratively, XMTNUM equals 3. Assuming an idle bus 12, and CPU 19a learns the availability of a message, then CPU 19a sets the XMITMODE flag. (See steps 200-209 of Fig. 10A).

[0034] An event (internal or external to MCU 18) causes CPU 19a to generate a message frame for transmission over bus 12. An application layer subroutine formats the message frame and stores the bytes in XMTBUF. XMTNUM gets set to the number of bytes that MCU 18 will transmit.

[0035] In this configuration, CPU 19a fills XMTBUF (usually an 11 byte buffer) backwards, as shown in Fig. 10C, so that the first byte to send gets stored as XMTBUF, XMTNUM-1 or XMTBUF, 2; the second byte stored as XMTBUF, XMTNUM-2 or XMTBUF, 1; and the third byte stored as XMTBUF, XMTNUM-3 or XMTBUF, 0.

[0036] After establishing XMTNUM=3, with bus 12 idle, CPU 19a sets XMTNDX=XMTNUM, illustratively to 3 and ARBNDX=XMTNUM or 3. ARBNDX is an index

into XMTBUF that points to the next byte coming back from bus 12. (See steps 210-212).

[0037] At step 214, CPU 19a receives instructions to load XMTNDX-1 or the data in XMTBUF, 2 into the SPI data register (SPDR) 19f. Then at step 216, CPU 19a sets BYT RDY pin HI. This informs link 22 of a byte ready for transmission. (See Fig. 7G, index #1).

[0038] CPU 19a then sets XMTNDX equal to XMTNDX-1 or in this illustration to 2. CPU 19a receives instructions to determine if the SPI-XFR between MCU 18 and Link 22 has been completed. (See Fig. 10A, step 219). Usually, this transfer consumes about 11 μ s well within the first 400 μ s polling cycle. The byte that was stored in the LSPDR 28, which is unusable, comes into the MSPDR 19f of MCU 18.

[0039] After completion of the transfer, the hardware in MCU 18 automatically clears the SPIF bit in the SPI Status Register, (SPSR) (not shown). Then CPU 19a receives instructions to load the byte received from the LSPDR 28 into a non-volatile MCU register. This completes the solicited transfer. MCU 18 does not care about the initial byte that comes back into the MSPDR from Link 22. (See steps 220-221).

[0040] If XMTNDX does not equal zero, CPU 19a sets BYT RDY pin LO. This means that there are more bytes to transmit. If there are more bytes to transmit, then CPU 19a loads the second byte to be transmitted in the MSPDR 19f by writing over the returned initial byte and then setting BYT RDY pin HI a second time. (See steps 222-224). The rising edge of the second short BYT RDY pulse must occur at least 2.1 μ s after the falling edge of the last SPISCK pulse of the SPI-XFR. (See the waveforms in Fig. 7D and 7F). When the SOF symbol ends, Link 22 begins transmitting the first byte onto bus 12. CPU 19a then sets XMTNDX equal to XMTNDX-1. XMTNDX is now 1. If XMTNDX is not equal to zero, then CPU 19a sets BYT RDY pin LOW. CPU 19a then exits the solicited transfer routine. (See steps 226-229).

MCU RECEIPT OF AN UNSOLICITED BYTE BY SPI XFR and SPI XFR OF 2ND BYTE

[0041] After loading byte 1 on bus 12, IDR 20 and Link 22 reflect byte 1 (received off bus 12) back to MCU 18 via an unsolicited transfer. (See Fig. 7G, index #3).

[0042] In Fig. 10A, during another 400 μ s polling cycle, at step 200, CPU 19a determines if SPIF is set. This serial peripheral transfer flag notifies MCU 18 that the solicited data transfer between MCU 18 and Link 22 has been completed. Hence the program branches to the unsolicited transfer program on Fig. 10B.

ARBITRATION CHECKED

[0043] At the same time MCU 18 receives back the unsolicited byte 1, Link 22 receives byte 2 (if any) from MCU 18 via the SPI-XFR. During the time Link 22 was attempting to transmit byte 1, Link 22 could have lost

arbitration to another transmitting node. If so, the byte it was attempting to transmit is not the byte that ends up being transmitted on bus 12. MCU 18 checks each byte it receives back from Link 22 against the byte it attempted to transmit. If byte 1 does not match, MCU 18 switches to a receive operation. If the first byte received back matches but subsequent bytes do not, MCU 18 assumes an error has occurred. This is due to the fact that this system utilizes a single-byte header and arbitration should be resolved with the first byte. If the second or subsequent bytes do not match, MCU 18 terminates transmission of the message and waits for the next idle bus period.

PERFORMANCE OF THE UNSOLICITED TRANSFER

[0044] In Fig. 10B, after the SPIF bit goes to a logic 1, MCU 18 receives instructions to clear BYT RDY and then to determine if XMITMODE flag is set. Since node 1 is transmitting, the program branches to the next instruction which requires determining if ARBNDX is equal to zero. If so, then the program exits but, illustratively, since ARBNDX was previously set to 3, CPU 19a reacts to the next instruction of accessing the SPI STATUS REGISTER (SPSR). Then CPU 19a receives the instruction to load the data byte read from bus 12 into a nonvolatile MCU register (not shown). This action also clears the SPIF bit.

[0045] Then CPU 19a determines if the byte received from bus 12 equals the byte transmitted. If so, CPU 19a updates the ARBNDX by setting it to [3-1] or 2. (See steps 230-240).

TRANSFER OF THE LAST BYTE

[0046] CPU 19a receives instructions to determine if XMTNDX equals zero. Since XMTNDX does not, CPU 19a executes the next instruction to place the last byte in the SPDR (XMTNDX=1). Then CPU 19a receives the instruction to set the BYT RDY pin HI. Leaving the BYT RDY pin HI indicates that the byte in the MSPDR is the last byte to be transmitted. CPU 19a then moves to execute the instruction to update the XMTNDX before exiting the program. XMTNDX now becomes zero.

[0047] Illustratively, CPU 19a loads the last byte into MSPDR 19f and asserts the BYT RDY pin HI. BYT RDY pin remains HI through the next unsolicited transfer. This is referred to as the long form of BYT RDY. This tells Link 22 that the byte sitting in MSPDR 19f is the last byte to transmit. As with the short BYT RDY pulse, the rising edge of a long BYT RDY must occur at least 2.1 μ s after the falling edge of the last SCK pulse of the SPI XFR. During the unsolicited transfer (Fig. 7G, index #4), MCU 18 receives BYTE 2, while Link 22 receives byte 3. Following the transfer, MCU 18 returns the BYT RDY pin LO.

[0048] Illustratively, two more unsolicited transfers must occur, namely an unsolicited transfer for the third

byte and one for the CRC byte. For each of these two unsolicited transfers, XMTNDX equals zero. So no more bytes are sent to Link 22 for transmission. After each unsolicited transfer, CPU 19a must check the byte received against XMTBUF, ARBNDX-1 and decrement ARBNDX afterwards.

LINK AUTOMATICALLY TRANSMITS A CRC

[0049] Following the transmission of byte 3, unsolicited transfer (Fig. 7G, index # 5) sends byte 3 to MCU 18. Then MCU 18 exits the Unsolicited program. Link 22 automatically transmits the proper CRC byte on to bus 12. Unsolicited transfer (index # 6) sends the CRC to MCU 18. During the unsolicited transfer for the CRC, ARBNDX equals 0, so MCU 18 does not participate in this transfer. The idle pin returns to LO (index # 7) indicating bus 12 is once again idle. This causes an interrupt to MCU 18.

[0050] Since other modifications and changes varied to fit particular operating requirements and environments will be apparent to those skilled in the art, the invention is not considered limited to the example chosen for purposes of disclosure, and covers all changes and modifications which do not constitute departures from the scope of this invention.

Claims

1. A link apparatus containing a circuit for interchanging data bytes using a serial peripheral interface circuit (19d) of a microcontroller (18a) at a chosen clock rate to accomplish loading a series of data bytes from the microcontroller into the link apparatus and unloading another series of data bytes from the link apparatus into the microcontroller substantially simultaneously and then for transferring each bit of the loaded byte onto a single-wire communications bus (12) as J1850 data symbols at the chosen clock rate but delayed in proportion to a variation in length of the variable pulse width of each symbol, the link apparatus being contained in a node (16a) of a multiple node (16a-f) vehicle communications network and having output terminals connected to the bus (12), the bus and the J1850 symbols being prescribed for use in an automobile industrial standard for data communications network interfaces, each node also having the microcontroller connected between an external input device (14a) and input terminals of the link apparatus for receiving a chosen measurand from the input device and then translating the measurand into digital byte messages in response to changes of the measurand and for cooperating with the link apparatus to affect serial peripheral interface transfers, the link apparatus being characterized by comprising:

a symbol encoder/decoder (23a, et al.) for receiving the bits of the bytes loaded from the microcontroller and then formatting each bit of the bytes into a symmetrical, variable pulse width modulated signal, the variable pulse width modulated signal being routed from the link apparatus to an integrated driver/receiver circuit that connects to the bus, the integrated driver/receiver circuit (20) containing a transmitter circuit for converting each variable pulse width modulated signal into a trapezoidal shaped waveform analog signal containing information expressing a magnitude value of the digital bit component of the byte message being communicated so that other nodes on the bus might use the communicated information, the integrated driver/receiver circuit (20) also containing a receiver circuit for reflecting each trapezoidal shaped waveform analog signal placed on the bus and reconvert-ing the trapezoidal shaped waveform analog signal into a restructured variable pulse width modulated bit, the restructured bit being routed back to the link device where the variable pulse width modulated bit is converted by the symbol encoder/decoder into a digital data bit, the digital data bits being then reassembled as a data byte in a data register of the link and then unloaded into the microcontroller via a serial peripheral interface transfer providing arbitration and status information to the microcontroller regarding the placement of the data message on the communications bus, the link apparatus further comprising:

A) A state machine device for establishing a series of operating states for the node comprising:

- 1) a statein unit (62a) having a plurality of synchronous and asynchronous holding registers;
- 2) a main state machine unit (48) having:

- a) a next state decoder (Fig. 9A) at an input port,
- b) a memory circuit composed of a parallel arrangement of plurality of flip-flops (Fig. 9A) having input terminals connected to the output of the next state decoder,
- c) the next state decoder having input terminals (Fig. 9A) connected to the output of the memory circuit, and
- d) synchronized output gates (Fig. 9A) having input terminals connected to the output of the memory circuit and output terminals of the statein unit for providing synchronous output state signals within the link apparatus,

the state machine device being used to monitor

output signals from the symbol encoder/decoder that provide information concerning the status of and the symbols on the bus and to provide a serial peripheral interface transfer clock and handshaking signals to the microcontroller, the state machine device also determining when the bus is idle and in a condition to receive a new message and to respond to handshaking signals from the microcontroller indicating that the microcontroller has converted measurand information into data bytes;

B) a stateout unit (70a) having input terminals connected to output terminals of the state machine device and the statein unit for providing control signals within the link apparatus and to the microcontroller;

C) a link apparatus serial peripheral interface data register circuit (28a) having an input connected to receive each data bit of the byte transferred from a microcontroller serial peripheral interface data register during a serial peripheral interface transfer operation of the microcontroller;

D) an encoder circuit (37a) for accepting a signal from the state machine device during a particular state and then automatically generating a coded address that routes to the symbol encoder/decoder for generating a variable pulse width modulated signal that routes to the transmitter circuit of the integrated driver/receiver for placing a start of frame protocol symbol on the bus (12) during a chosen state of the state machine device and for accepting data bits from the link apparatus serial peripheral interface data register circuit (28a) in response to handshaking signals from the state machine device and control signals of the symbol encoder/decoder in order to initiate the loading of data symbols on the bus (12) after the start of frame protocol symbol has been communicated over the bus;

E) a firmware program being stored in a ROM (19b) of the microcontroller (18a) that instructs a central processing unit to perform a solicited transfer of data bytes in a message frame and for responding to unsolicited transfers of data bytes interchanged with the link apparatus (22a); and

F) an arbitration determining device connected to the symbol encoder/decoder and the main state machine unit containing circuits for determining arbitration of the start of frame protocol symbol with respect to other start of frame protocol symbols that might appear on the bus nearly simultaneously, the arbitration determining device providing an error signal that resets the state machine device to an initial state if another protocol symbol from another node ap-

pears on the bus prior to a transmitting node.

2. The link apparatus of claim 1, wherein the symbol encoder/decoder receives a reference clock signal from the microcontroller and divides that signal in half forming a symbol encoder/decoder reference clock signal, wherein the symbol encoder/decoder provides a circuit that offsets the symbol encoder/decoder reference clock signal by 1/4 of a cycle to form a further signal and a serial peripheral interface clock signal, wherein the further signal is used to synchronize the output gates of the main state machine unit and the serial peripheral interface clock signal is used by the microcontroller to clock the serial peripheral interface transfers, wherein the symbol encoder/decoder provides a circuit combining the symbol encoder/decoder reference clock signal and the further signal to form a pulse $\frac{1}{2}$ the frequency of the reference clock signal.
3. The link apparatus of claim 2, wherein the statein unit includes circuits that combine certain output signals of the memory unit of the main state machine unit with the further signal forming a shift clock signal that clocks the link apparatus serial peripheral interface data register circuit during unloading bits during serial peripheral interface transfer.
4. The link apparatus of claim 3, wherein an analog-to-digital converter circuit within the microcontroller receives measurands from the input device and converts the variations of the measurands into a series of digital byte messages, wherein the central processing unit polls the firmware to determine whether the state machine device has provided a handshaking signal to the microcontroller informing the microcontroller of the condition of the bus, wherein if the bus is idle, and the microcontroller has a byte ready to include in a message, then the central processing unit, during a polling cycle of the firmware asserts a short handshaking signal to the link apparatus to inform the link apparatus that a byte is ready.
5. The link apparatus of claim 4, wherein the microcontroller initiates a solicited serial peripheral interface transfer wherein the microcontroller exchanges a header byte from the microcontroller serial peripheral interface data register circuit with an unknown byte from the link apparatus serial peripheral interface data register, wherein the serial peripheral interface clock signal clocks the serial peripheral interface transfer.
6. The link apparatus of claim 5, wherein after the occurrence of the serial peripheral interface transfer, the link apparatus automatically places a start of frame protocol symbol on the bus, wherein after de-

terminating that the start of frame symbol won arbitration using the arbitration determining circuit, the link apparatus unloads each bit of the header byte as a variable pulse width modulated bit to the integrated driver/receiver clocked by the shift clock signal, wherein the integrated driver/receiver changes the variable pulse width modulated signal into J1850 symbols.

7. The link apparatus of claim 6, wherein each reflected bit from the integrated driver/receiver is checked to determine if arbitration occurs with any bit on the bus from a competing node, wherein each arbitrated bit is restructured into a byte and stored in the link apparatus serial peripheral interface data register circuit for an unsolicited transfer exchange with the next data byte stored in the microcontroller serial peripheral interface data register, wherein the microcontroller conducts unsolicited serial peripheral interface transfers with all succeeding bytes until the last byte is loaded in the microcontroller serial peripheral interface data register.
8. The link apparatus of claim 7, wherein the firmware program in the microcontroller instructs the central processing unit to assert a long handshaking signal to the link apparatus if the last byte is loaded into the microcontroller serial peripheral interface data register and wherein if any of the data bytes after the header byte do not match during arbitration, the microcontroller asserts a control bit and waits for the next idle handshake signal from the link apparatus.

Patentansprüche

1. Verbindungsvorrichtung mit einer Schaltung für den Austausch von Datenbytes unter Verwendung einer seriellen peripheren Schnittstellenschaltung (19d) eines Mikrocontrollers (18a) mit einer ausgewählten Taktrate, um das Laden einer Reihe von Datenbytes aus dem Mikrocontroller in die Verbindungsvorrichtung zu erreichen sowie im wesentlichen gleichzeitig das Entladen einer weiteren Reihe von Datenbytes von der Verbindungsvorrichtung in den Mikrocontroller sowie zur folgenden Übertragung jedes Bits des geladenen Bytes auf einen Einleitungs-Kommunikationsbus (12) als J1850-Datensymbole mit der ausgewählten Taktrate, jedoch verzögert im Verhältnis zu einer Veränderung der Länge der variablen Impulsbreite jedes Symbols, wobei sich die Verbindungsvorrichtung in einem Knoten (16a) eines Fahrzeug-Kommunikationsnetzes mit mehreren Knoten (16a-f) befindet, und wobei die Ausgangsanschlüsse mit dem Bus (12) verbunden sind, wobei der Bus und die J1850-Symbole zur Verwendung gemäß einer Industrienorm für Kraftfahrzeuge für Schnittstellen von Datenkommunikationsnetzen vorgeschrieben sind, wobei jeder Knoten ferner den Mikrocontroller aufweist, der zwischen einer externen Eingabevorrichtung (14a) und die Eingangsanschlüsse der Verbindungsvorrichtung geschaltet ist, um einen ausgewählten Meßwert von der Eingabevorrichtung zu empfangen und um den Meßwert als Reaktion auf Änderungen des Meßwertes in digitale Bytesnachrichten umzuwandeln, und um mit der Verbindungsvorrichtung so zusammenzuwirken, daß serielle periphere Schnittstellenübertragungen bewirkt werden, wobei die Verbindungsvorrichtung dadurch gekennzeichnet ist, daß sie folgendes umfaßt:
einen Symbol-Codierer/Decodierer (23a, et al) zum Empfang der Bits der von dem Mikrocontroller geladenen Bytes und zum folgenden Formatieren jedes Bits der Bytes in ein symmetrisches modulierte Signal mit variabler Impulsbreite, wobei das modulierte Signal mit variabler Impulsbreite von der Verbindungsvorrichtung zu einer integrierten Treiber/Empfänger-Schaltung geleitet wird, welche eine Verbindung mit dem Bus vorsieht, wobei die integrierte Treiber-/Empfänger-Schaltung (20) eine Senderschaltung aufweist, die dazu dient, jedes modulierte Signal mit variabler Impulsbreite in ein analoges trapezförmiges Signal umzuwandeln, das Daten bzw. Informationen aufweist, die einen Größenwert der digitalen Bitkomponente der Bytesnachricht ausdrückt, die übertragen wird, so daß andere Knoten an dem Bus die übertragenen Informationen nutzen können, wobei die integrierte Treiber/Empfänger-Schaltung (20) ferner eine Empfängerschaltung aufweist, die dazu dient, jedes analoge trapezförmige Signal zu reflektieren, das auf dem Bus platziert wird, und wobei das trapezförmige analoge Signal in ein restrukturiertes modulierte Bit mit variabler Impulsbreite zurück umgewandelt wird, wobei das restrukturierte Bit zurück zu der Verbindungsvorrichtung geführt wird, wo das modulierte Bit mit variabler Impulsbreite durch den Symbol-Codierer/Decodierer in ein digitales Datenbit umgewandelt wird, wobei die digitalen Datenbits danach als Datenbyte in einem Datenregister der Verbindungsvorrichtung erneut zusammengesetzt und danach per serielle periphere Schnittstellenübertragung in den Mikrocontroller entladen werden, wobei der Mikrocontroller mit Arbitrations- und Statusinformationen bezüglich der Platzierung der Datennachrichten auf dem Kommunikationsbus versorgt wird, wobei die Verbindungsvorrichtung ferner folgendes umfaßt:
A) eine Zustands-Maschinenvorrichtung zur Erzeugung einer Reihe von Betriebszuständen für den Knoten, die folgendes umfaßt:
1) eine Zustand-Ein-Einheit (62a) mit einer Mehrzahl von synchronen und asynchrone

tionsnetzen vorgeschrieben sind, wobei jeder Knoten ferner den Mikrocontroller aufweist, der zwischen einer externen Eingabevorrichtung (14a) und die Eingangsanschlüsse der Verbindungsvorrichtung geschaltet ist, um einen ausgewählten Meßwert von der Eingabevorrichtung zu empfangen und um den Meßwert als Reaktion auf Änderungen des Meßwertes in digitale Bytesnachrichten umzuwandeln, und um mit der Verbindungsvorrichtung so zusammenzuwirken, daß serielle periphere Schnittstellenübertragungen bewirkt werden, wobei die Verbindungsvorrichtung dadurch gekennzeichnet ist, daß sie folgendes umfaßt:

einen Symbol-Codierer/Decodierer (23a, et al) zum Empfang der Bits der von dem Mikrocontroller geladenen Bytes und zum folgenden Formatieren jedes Bits der Bytes in ein symmetrisches modulierte Signal mit variabler Impulsbreite, wobei das modulierte Signal mit variabler Impulsbreite von der Verbindungsvorrichtung zu einer integrierten Treiber/Empfänger-Schaltung geleitet wird, welche eine Verbindung mit dem Bus vorsieht, wobei die integrierte Treiber-/Empfänger-Schaltung (20) eine Senderschaltung aufweist, die dazu dient, jedes modulierte Signal mit variabler Impulsbreite in ein analoges trapezförmiges Signal umzuwandeln, das Daten bzw. Informationen aufweist, die einen Größenwert der digitalen Bitkomponente der Bytesnachricht ausdrückt, die übertragen wird, so daß andere Knoten an dem Bus die übertragenen Informationen nutzen können, wobei die integrierte Treiber/Empfänger-Schaltung (20) ferner eine Empfängerschaltung aufweist, die dazu dient, jedes analoge trapezförmige Signal zu reflektieren, das auf dem Bus platziert wird, und wobei das trapezförmige analoge Signal in ein restrukturiertes modulierte Bit mit variabler Impulsbreite zurück umgewandelt wird, wobei das restrukturierte Bit zurück zu der Verbindungsvorrichtung geführt wird, wo das modulierte Bit mit variabler Impulsbreite durch den Symbol-Codierer/Decodierer in ein digitales Datenbit umgewandelt wird, wobei die digitalen Datenbits danach als Datenbyte in einem Datenregister der Verbindungsvorrichtung erneut zusammengesetzt und danach per serielle periphere Schnittstellenübertragung in den Mikrocontroller entladen werden, wobei der Mikrocontroller mit Arbitrations- und Statusinformationen bezüglich der Platzierung der Datennachrichten auf dem Kommunikationsbus versorgt wird, wobei die Verbindungsvorrichtung ferner folgendes umfaßt:

A) eine Zustands-Maschinenvorrichtung zur Erzeugung einer Reihe von Betriebszuständen für den Knoten, die folgendes umfaßt:

- 1) eine Zustand-Ein-Einheit (62a) mit einer Mehrzahl von synchronen und asynchrone

nen Haltereinheit;

2) eine Hauptzustands-Maschineneinheit (48), mit:

- a) einem Decodierer (Figur 9A) für den nächsten Zustand an einem Eingangsanschluß; 5
- b) einer Speicherschaltung, die sich aus einer parallelen Anordnung einer Mehrzahl von Flip-flops (Figur 9A) zusammensetzt, wobei die Eingangsanschlüsse mit dem Ausgang des Decodierers für den nächsten Zustand verbunden sind; 10
- c) wobei der Decodierer für den nächsten Zustand Eingangsanschlüsse (Figur 9A) aufweist, die mit dem Ausgang der Speicherschaltung verbunden sind; und 15
- d) synchronisierten Ausgabegattern (Figur 9A), deren Eingangsanschlüsse mit dem Ausgang der Speicherschaltung und den Ausgangsanschlüssen der Zustand-Ein-Einheit verbunden sind, um synchrone Ausgangszustandssignale in der Verbindungsvorrichtung vorzusehen; 20 25

wobei die Zustands-Maschinenvorrichtung zur Überwachung der Ausgangssignale von dem Symbol-Codierer/Decodierer verwendet wird, die Informationen über den Zustand und die Symbole auf dem Bus vorsehen, und um einen seriellen peripheren Schnittstellen-Übernahmetakt und Quittungssignale für den Mikrocontroller vorzusehen, wobei die Zustands-Maschinenvorrichtung ferner bestimmt, wenn sich der Bus im Ruhezustand und in einem Zustand befindet, in dem er eine neue Nachricht empfangen und auf Quittungssignale von dem Mikrocontroller reagieren kann, die anzeigen, daß der Mikrocontroller Meßwertinformationen in Datenbytes umgewandelt hat; 30 35 40 45

B) eine Zustand-Aus-Einheit (70a) mit Eingangsanschlüssen, die mit Ausgangsanschlüssen der Zustands-Maschinenvorrichtung und der Zustand-Ein-Einheit verbunden sind, um Steuersignale in der Verbindungsvorrichtung und an den Mikrocontroller vorzusehen; 50

C) eine Datenregisterschaltung (28a) der seriellen peripheren Schnittstelle der Verbindungsvorrichtung, wobei ein Eingang so angeschlossen ist, daß er jedes Datenbit von dem Byte empfängt, das während einem Übertragungsbetrieb des Mikrocontrollers der seriellen peripheren Schnittstelle von einem Datenregister der seriellen peripheren Schnittstelle des Mi-

krocontrollers übertragen wird;

D) eine Codierschaltung (37a) zur Annahme eines Signals von der Zustands-Maschinenvorrichtung während einem bestimmten Zustand, und wobei danach automatisch eine codierte Adresse erzeugt wird, die zu dem Symbol-Codierer/Decodierer führt, um ein moduliertes Signal mit variabler Impulsbreite zu erzeugen, das zu der Senderschaltung des integrierten Treibers/Empfängers geleitet wird, um während einem ausgewählten Zustand der Zustands-Maschinenvorrichtung einen Beginn eines Rahmenprotokollsymbols an dem Bus (12) zu platzieren, und um als Reaktion auf Quittungssignale von der Zustands-Maschinenvorrichtung und Steuersignale von dem Symbol-Codierer/Decodierer Datenbits von der Datenregisterschaltung (28a) der seriellen peripheren Schnittstelle der Verbindungsvorrichtung anzunehmen, um das Laden der Datensymbole auf dem Bus (12) einzuleiten, nach dem der Beginn des Rahmenprotokollsymbols über den Bus übertragen worden ist;

E) ein Firmware-Programm, das in einem ROM-Speicher (19b) des Mikrocontrollers (18a) gespeichert ist, das eine Zentraleinheit anweist eine abgerufene Übertragung von Datenbytes in einem Nachrichtenrahmen auszuführen, und wobei es ferner dazu dient, auf frei laufende Übertragungen von Datenbytes zu reagieren, die mit der Verbindungsvorrichtung (22a) ausgetauscht werden; und

F) eine Arbitrations-Bestimmungsvorrichtung, die mit dem Symbol-Codierer/Decodierer und der Hauptzustands-Maschineneinheit verbunden ist, welche Schaltungen für die Bestimmung einer Arbitration des Beginns eines Rahmenprotokollsymbols in Bezug auf andere Beginne von Rahmenprotokollsymbolen aufweist, die nahezu gleichzeitig auf dem Bus auftreten können, wobei die Arbitrations-Bestimmungsvorrichtung ein Fehlersignal vorsieht, das die Zustands-Maschinenvorrichtung in einen Ausgangszustand zurücksetzt, wenn ein anderes Protokollsymbol von einem anderen Knoten vor einem sendenden Knoten auf dem Bus auftritt.

2. Verbindungsvorrichtung nach Anspruch 1, wobei der Symbol-Codierer/Decodierer ein Bezugstaktssignal von dem Mikrocontroller empfängt und dieses Signal durch Division halbiert, wobei ein Symbol-Codierer/Decodierer-Bezugstaktssignal gebildet wird, wobei der Symbol-Codierer/Decodierer eine Schaltung vorsieht, die das Symbol-Codierer/Decodierer-Bezugstaktssignal um 1/4 eines Zyklus versetzt, um ein weiteres Signal sowie ein Taktssignal für die serielle periphere Schnittstelle zu bilden, wo-

bei das weitere Signal dazu verwendet wird, die Ausgabegatter der Hauptzustands-Maschineneinheit zu synchronisieren, und wobei das Taktsignal für die serielle periphere Schnittstelle von dem Mikrocontroller dazu verwendet wird, die Übertragungen der seriellen peripheren Schnittstelle zu takten, wobei der Symbol-Codierer/Decodierer eine Schaltung vorsieht, welche das Bezugstaktsignal des Symbol-Codierers/Decodierers und das weitere Signal verknüpft, so daß ein Impuls mit der Hälfte der Frequenz des Bezugstaktsignals gebildet wird.

3. Verbindungsvorrichtung nach Anspruch 2, wobei die Zustand-Ein-Einheit Schaltungen umfaßt, die bestimmte Ausgangssignale der Speichereinheit der Hauptzustands-Maschineneinheit mit dem weiteren Signal verknüpfen, wobei ein Schiebetaktsignal gebildet wird, das die Datenregisterschaltung der seriellen peripheren Schnittstelle der Verbindungsvorrichtung während dem Entladen der Bits während der Übertragung der seriellen peripheren Schnittstelle taktet.
4. Verbindungsvorrichtung nach Anspruch 3, wobei eine Analog-Digital-Umsetzerschaltung in dem Mikrocontroller Meßwerte von der Eingabevorrichtung empfängt und Abweichungen der Meßwerte in eine Reihe digitaler Bytenachrichten umwandelt, wobei die Zentraleinheit die Firmware abrufen, um festzustellen, ob die Zustands-Maschinenvorrichtung ein Quittungssignal an den Mikrocontroller vorgesehen hat, das dem Mikrocontroller den Zustand des Busses mitteilt, wobei der Mikrocontroller für den Fall, daß sich der Bus im Ruhezustand befindet, ein bereits Byte zum Einfügen in eine Nachricht aufweist, wobei die Zentraleinheit danach während einem Abrufzyklus der Firmware ein kurzes Quittungssignal an die Verbindungseinrichtung vorsieht, um die Verbindungsvorrichtung darüber zu informieren, daß ein Byte bereit ist.
5. Verbindungsvorrichtung nach Anspruch 4, wobei der Mikrocontroller eine abgerufene Übertragung der seriellen peripheren Schnittstelle einleitet, wobei der Mikrocontroller ein Kopfbyte von der Datenregisterschaltung der seriellen peripheren Schnittstelle des Mikrocontrollers mit einem unbekannten Byte des Datenregisters der seriellen peripheren Schnittstelle der Verbindungsvorrichtung austauscht, wobei das Taktsignal der seriellen peripheren Schnittstelle die Übertragung der seriellen peripheren Schnittstelle taktet.
6. Verbindungsvorrichtung nach Anspruch 5, wobei die Verbindungsvorrichtung nach dem Auftreten der Übertragung der seriellen peripheren Schnittstelle automatisch einen Beginn des Rahmenprotokollsymbols auf dem Bus platziert, wobei die Ver-

bindungsvorrichtung nach der Feststellung, daß der Beginn des Rahmensymbols Arbitration unter Verwendung der Arbitrations-Bestimmungsschaltung vorgesehen hat, jedes Bit des Kopfbytes als moduliertes Bit mit variabler Impulsbreite in den integrierten Treiber/Empfänger entlädt, der durch das Schiebetaktsignal getaktet wird, wobei der integrierte Treiber/Empfänger das modulierte Signal mit variabler Impulsbreite in J1850-Symbole umwandelt.

7. Verbindungsvorrichtung nach Anspruch 6, wobei jedes von dem integrierten Treiber/Empfänger reflektierte Bit geprüft wird, um festzustellen, ob eine Arbitration mit einem anderen Bit auf dem Bus eines konkurrierenden Knotens stattfindet, wobei jedes arbitrierte Bit in ein Byte neu strukturiert und in der Datenregisterschaltung der seriellen peripheren Schnittstelle der Verbindungsvorrichtung für einen freilaufenden Übertragungsaustausch mit dem als nächstes in dem Datenregister der seriellen peripheren Schnittstelle des Mikrocontrollers gespeichert wird, wobei der Mikrocontroller freilaufende Übertragungen der seriellen peripheren Schnittstelle mit allen folgenden Bytes durchführt, bis das letzte Byte in das Datenregister der seriellen peripheren Schnittstelle des Mikrocontrollers geladen wird.
8. Verbindungsvorrichtung nach Anspruch 7, wobei das Firmware-Programm in dem Mikrocontroller die Zentraleinheit anweist, ein langes Quittungssignal an die Verbindungseinrichtung vorzusehen, wenn das letzte Byte in das Datenregister der seriellen peripheren Schnittstelle des Mikrocontrollers geladen wird, und wobei für den Fall, daß etwaige Datenbytes nach dem Kopfbyte während der Arbitration nicht übereinstimmen, der Mikrocontroller ein Steuerbit vorsieht und auf das nächste Ruhezustands-Quittungssignal von der Verbindungsvorrichtung wartet.

Revendications

1. Appareil de liaison contenant un circuit d'échange d'octets de données à l'aide d'un circuit d'interface périphérique série (19d) d'un organe à microcontrôleur (18a) à une fréquence choisie d'horloge pour assurer le chargement d'une série d'octets de données du microcontrôleur à l'appareil de liaison et le déchargement d'une autre série d'octets de données de l'appareil de liaison au microcontrôleur de manière pratiquement simultanée, puis pour transférer chaque bit d'octet chargé dans un bus (12) de communications à un seul fil sous forme de symboles de données J1850 à une fréquence choisie d'horloge, mais avec un retard proportionnel à la variation de longueur de la largeur d'impulsion varia-

ble de chaque symbole, l'appareil de liaison étant contenu dans un noeud (16a) d'un réseau de communications de véhicule à plusieurs noeuds (16a-f) et ayant des bornes de sortie connectées au bus (12), le bus et les symboles J1850 étant destinés à être utilisés selon une norme industrielle d'interface de réseau de communications de données pour automobiles, chaque noeud ayant aussi deux microcontrôleurs connectés entre un dispositif externe d'entrée (14a) et des bornes d'entrée de l'appareil de liaison afin qu'il reçoive un mesurande choisi du dispositif d'entrée puis transforme le mesurande en message numérique d'octets en fonction des changements du mesurande, et étant destiné à coopérer avec l'appareil de liaison pour assurer des transferts d'interface périphérique série, l'appareil de liaison étant caractérisé en ce qu'il comprend :

un codeur-décodeur de symboles (23a, etc.) destiné à recevoir les bits des octets chargés depuis le microcontrôleur, puis à formater chaque bit des octets en un signal symétrique modulé par impulsions de largeur variable, le signal modulé par impulsions de largeur variable étant acheminé de l'appareil de liaison à un circuit de pilotage et récepteur intégré qui est connecté au bus, le circuit (20) de pilotage et récepteur intégré contenant un circuit transmetteur destiné à transformer chaque signal modulé par impulsions de largeur variable en un signal analogique de forme d'onde trapézoïdale contenant des informations exprimant une valeur d'amplitude de la composante numérique de bit du message d'octets communiqué afin que les autres noeuds du bus puissent utiliser les informations communiquées, le circuit (20) de pilotage et récepteur intégré contenant aussi un circuit récepteur destiné à réfléchir chaque signal analogique de forme d'onde trapézoïdale placé sur le bus et à retransformer le signal analogique de forme d'onde trapézoïdale en un bit restructuré modulé par impulsions de largeur variable, le bit restructuré étant réacheminé vers le dispositif de liaison dans lequel le bit modulé par impulsions de largeur variable est transformé par le codeur-décodeur de symboles en un bit numérique de données, les bits numériques de données étant alors réassemblés sous forme d'un octet de données dans un registre de données de la liaison, puis étant déchargés dans le microcontrôleur par l'intermédiaire d'un transfert d'interface périphérique série transmettant des informations d'arbitrage et d'état au microcontrôleur, concernant la disposition du message de données dans le bus de communications, l'appareil de liaison comprenant en outre :

A) un automate fini destiné à établir une série d'états de fonctionnement du noeud, comprenant

1) une unité (62a) d'entrée d'état ayant plusieurs registres synchrones et asynchrones de maintien, et
2) une unité (48) à automate fini principal possédant :

- a) un décodeur d'état suivant (figure 9A) à une voie d'entrée,
- b) un circuit de mémoire composé d'un arrangement parallèle de plusieurs bascules (figure 9A) ayant des bornes d'entrée connectées à la sortie du décodeur d'état suivant,
- c) le décodeur d'état suivant ayant des bornes d'entrée (figure 9A) connectées à la sortie du circuit de mémoire, et
- d) des portes synchronisées de sortie (figure 9A) ayant des bornes d'entrée connectées à la sortie du circuit de mémoire et des bornes de sortie de l'unité d'entrée d'état destinées à transmettre des signaux synchrones d'état de sortie dans l'appareil de liaison,

le dispositif à automate fini étant utilisé pour contrôler les signaux de sortie du codeur-décodeur de symboles qui donne des informations concernant l'état du bus et les symboles dans le bus et pour donner une horloge de transfert d'interface périphérique série et des signaux d'établissement de liaison au microcontrôleur, le dispositif à automate fini déterminant aussi le moment où le bus est libre et à un état de réception d'un nouveau message et de réponse aux signaux d'établissement de liaison provenant du microcontrôleur et indiquant que le microcontrôleur a transformé l'information de mesurande en octets de données,

B) une unité (70a) de sortie d'état ayant des bornes d'entrée connectées aux bornes de sortie du dispositif à automate fini et de l'unité d'entrée d'état afin qu'elle transmette des signaux de commande dans l'appareil de liaison et au microcontrôleur,

C) un circuit (28a) à registre de données d'interface périphérique série d'appareil de liaison, ayant une entrée connectée pour la réception de chaque bit de données de l'octet transféré d'un registre de données d'interface périphérique série du microcontrôleur pendant une opération de transfert d'interface périphérique série du microcontrôleur,

D) un circuit codeur (37a) destiné à accepter un signal provenant du dispositif à automate fini pendant un état particulier, puis à créer automatiquement une adresse codée acheminée vers le codeur-décodeur de symboles pour la

création d'un signal modulé par impulsions de largeur variable qui achemine vers le circuit transmetteur du circuit de pilotage et récepteur intégré afin qu'un symbole de protocole de début de trame soit placé sur le bus (12) pendant un état choisi du dispositif à automate fini et pour l'acceptation de bits de données provenant du circuit (28a) à registre de données d'interface périphérique série de l'appareil de liaison en fonction des signaux d'établissement de liaison provenant du dispositif à automate fini et des signaux de commande du codeur-décodeur de symboles afin que le chargement de symboles de données sur le bus (12) soit déclenché après que le début du symbole de protocole de trame a été communiqué par le bus,

E) un programme de microinstructions étant mémorisé dans une mémoire morte ROM (19b) du microcontrôleur (18a) et donnant des instructions à une unité centrale de traitement pour l'exécution d'un transfert sollicité d'octets de données dans une trame de message et pour la réponse à des transferts non sollicités d'octets de données échangés avec l'appareil de liaison (22a), et

F) un dispositif de détermination d'arbitrage connecté au codeur-décodeur de symboles et à l'unité à automate fini contenant des circuits de détermination d'arbitrage du lancement d'un symbole de protocole de trame par rapport à d'autres lancements de symboles de protocole de trame qui peuvent apparaître presque simultanément dans le bus, le dispositif de détermination d'arbitrage donnant un signal d'erreur qui rétablit le dispositif à automate fini à un état initial si un autre symbole de protocole provenant d'un autre noeud apparaît dans le bus avant un noeud de transmission.

2. Appareil de liaison selon la revendication 1, dans lequel le codeur-décodeur de symbole reçoit un signal d'horloge de référence du microcontrôleur et divise ce signal en deux en formant un signal d'horloge de référence de codeur-décodeur de symboles, dans lequel le codeur-décodeur de symboles forme un circuit qui décale le signal d'horloge de référence de codeur-décodeur de symboles de 1/4 cycle pour former un signal supplémentaire et un signal d'horloge d'interface périphérique série, dans lequel le signal supplémentaire est utilisé pour synchroniser les portes de sortie de l'unité à automate fini et le signal d'horloge d'interface périphérique série est utilisé par le microcontrôleur pour commander les transferts d'interface périphérique série, et dans lequel le codeur-décodeur de symboles forme un circuit combinant le signal d'horloge de référence de codeur-décodeur de symboles et

le signal supplémentaire pour former une impulsion à la moitié de la fréquence du signal d'horloge de référence.

3. Appareil de liaison selon la revendication 2, dans lequel l'unité d'entrée d'état comprend des circuits qui combinent certains signaux de sortie de l'unité de mémoire de l'unité à automate fini au signal supplémentaire en formant un signal d'horloge de décalage qui commande le circuit à registre de données d'interface périphérique série de l'appareil de liaison lors du déchargement des bits au cours d'un transfert d'interface périphérique série.
4. Appareil de liaison selon la revendication 3, dans lequel un circuit convertisseur analogique-numérique placé dans le microcontrôleur reçoit des mesurandes du dispositif d'entrée et transforme les variations des mesurandes en une série de messages numériques d'octets, dans lequel l'unité centrale de traitement interroge les microinstructions pour déterminer si le dispositif à automate fini a donné un signal d'établissement de liaison au microcontrôleur, informant le microcontrôleur de l'état du bus, dans lequel, si le bus est libre et le microcontrôleur a un octet prêt à incorporer à un message, l'unité centrale de traitement, pendant le cycle d'interrogation des microinstructions, affecte un court signal d'établissement de connexion à l'appareil de liaison pour informer l'appareil de liaison qu'un octet est prêt.
5. Appareil de liaison selon la revendication 4, dans lequel le microcontrôleur déclenche un transfert d'interface périphérique série sollicité dans lequel le microcontrôleur échange un octet d'en-tête provenant du circuit à registre de données d'interface périphérique série du microcontrôleur avec un octet inconnu provenant du registre de données d'interface périphérique série de l'appareil de liaison, le signal d'horloge d'interface périphérique série commandant le transfert d'interface périphérique série.
6. Appareil de liaison selon la revendication 5, dans lequel, après le transfert d'interface périphérique série, l'appareil de liaison place automatiquement un symbole de protocole de début de trame dans le bus, et, après détermination du fait qu'un symbole de début de trame a été sélectionné pendant l'arbitrage réalisé par le circuit de détermination d'arbitrage, l'appareil de liaison décharge chaque bit de l'octet d'en-tête comme bit modulé par impulsions de largeur variable vers le circuit de pilotage et récepteur intégré commandé par le signal de d'horloge de décalage, le circuit de pilotage et récepteur intégré transformant le signal modulé par impulsions de largeur variable en symbole J1850.

7. Appareil de liaison selon la revendication 6, dans lequel chaque bit réfléchi par le circuit de pilotage et récepteur intégré est vérifié pour la détermination du fait qu'un arbitrage s'est produit avec un bit quelconque du bus à partir d'un noeud en concurrence, dans lequel chaque bit arbitré est restructuré dans un octet et est mémorisé dans le circuit à registre de données d'interface périphérique série d'appareil de liaison pour un échange par transfert non sollicité avec l'octet suivant de données conservé dans le registre de données d'interface périphérique série du microcontrôleur, et dans lequel le microcontrôleur effectue des transferts non sollicités d'interface périphérique série avec tous les octets suivants jusqu'à ce que le dernier octet soit chargé dans le registre de données d'interface périphérique série du microcontrôleur.
8. Appareil de liaison selon la revendication 7, dans lequel le programme formé des microinstructions contenues dans le microcontrôleur indique à l'unité centrale de traitement d'affecter un long signal d'établissement de communication à l'appareil de liaison lorsque le dernier octet est chargé dans le registre de donnée d'interface périphérique série du microcontrôleur, et dans lequel, si l'un quelconque des octets de données suivant l'octet d'en-tête ne correspond pas en cours d'arbitrage, le microcontrôleur affecte un bit de commande et attend le signal suivant d'établissement de communication libre provenant de l'appareil de liaison.

5

10

15

20

25

30

35

40

45

50

55

14

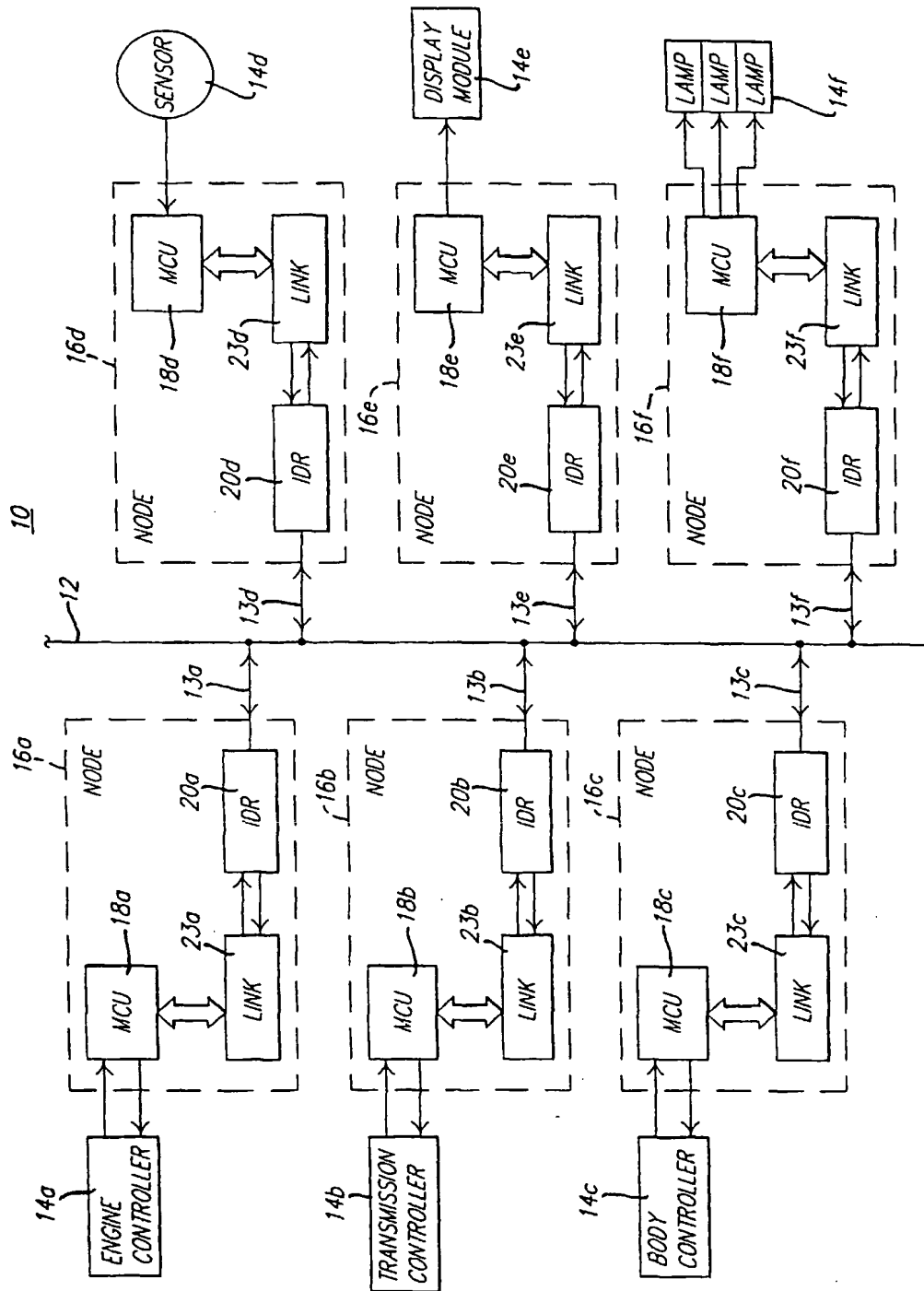
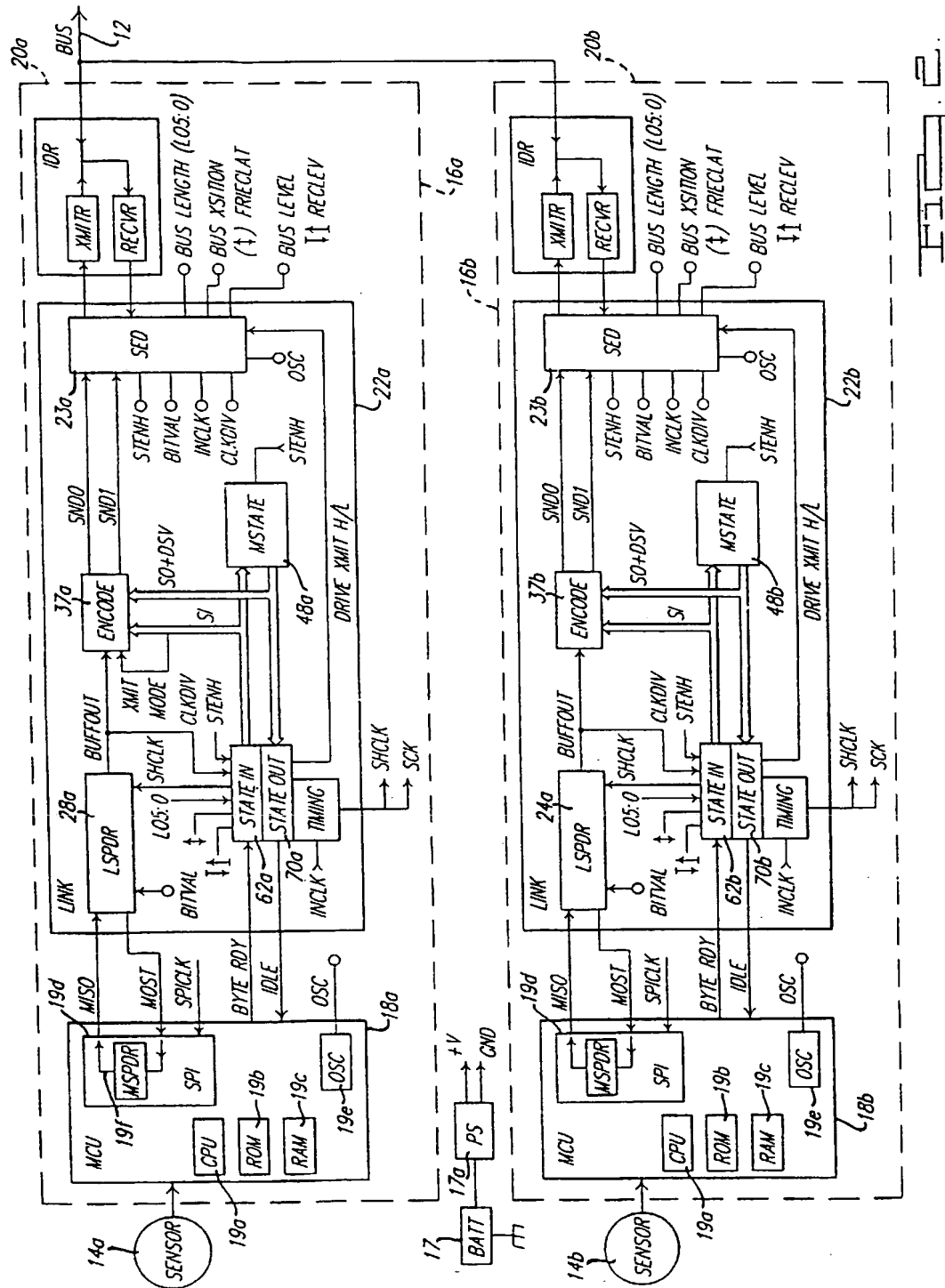
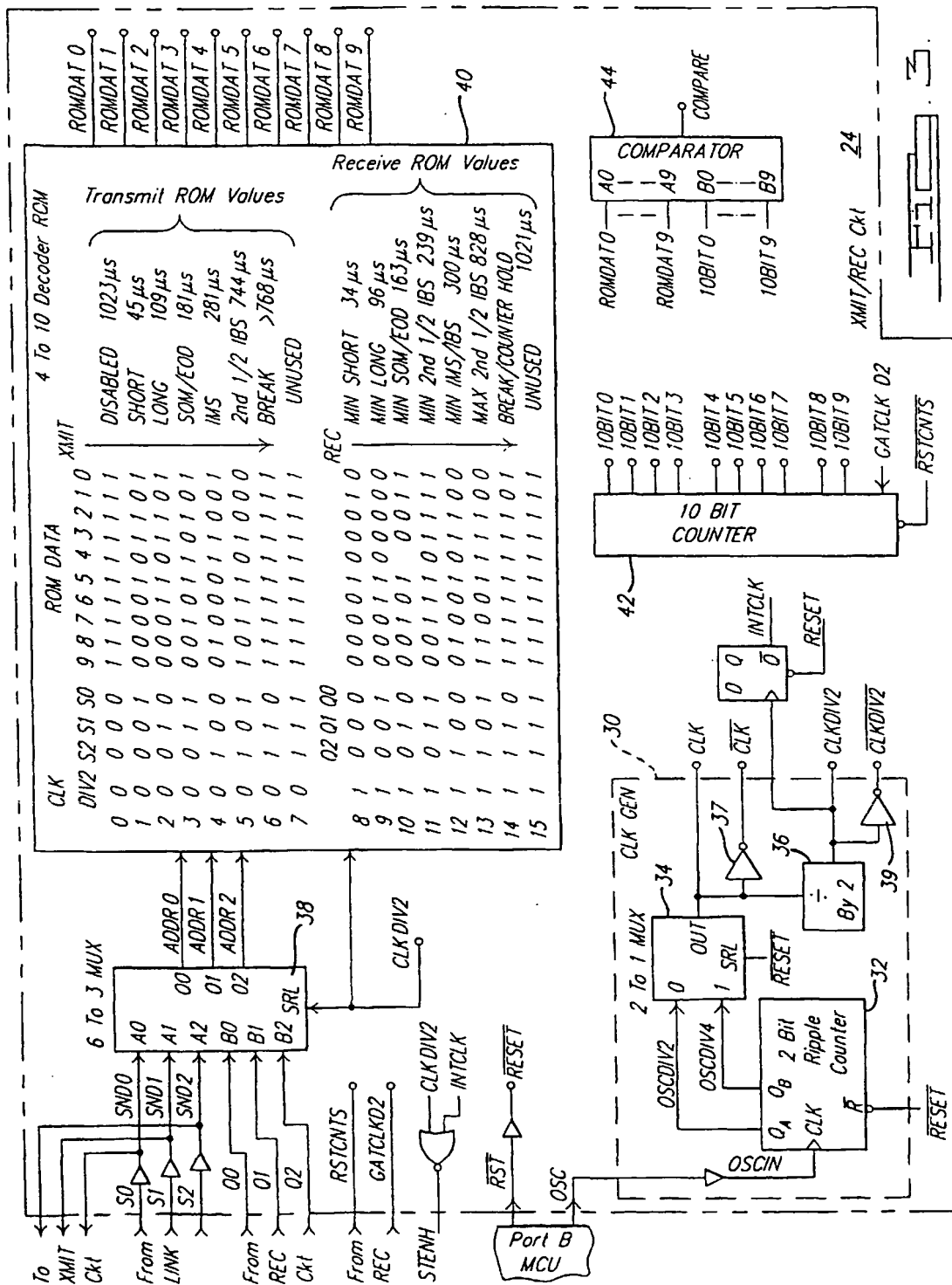
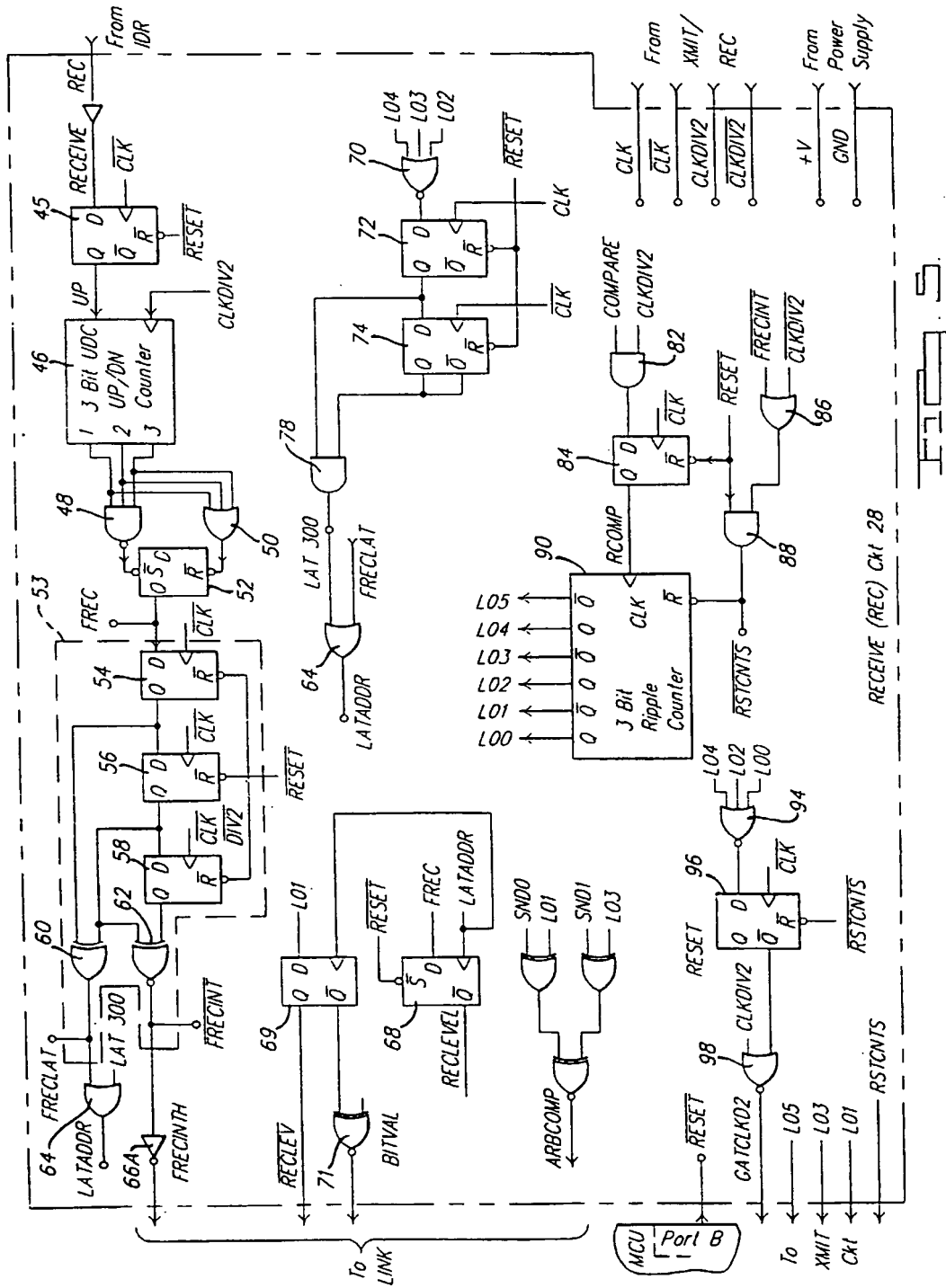


FIG. 1









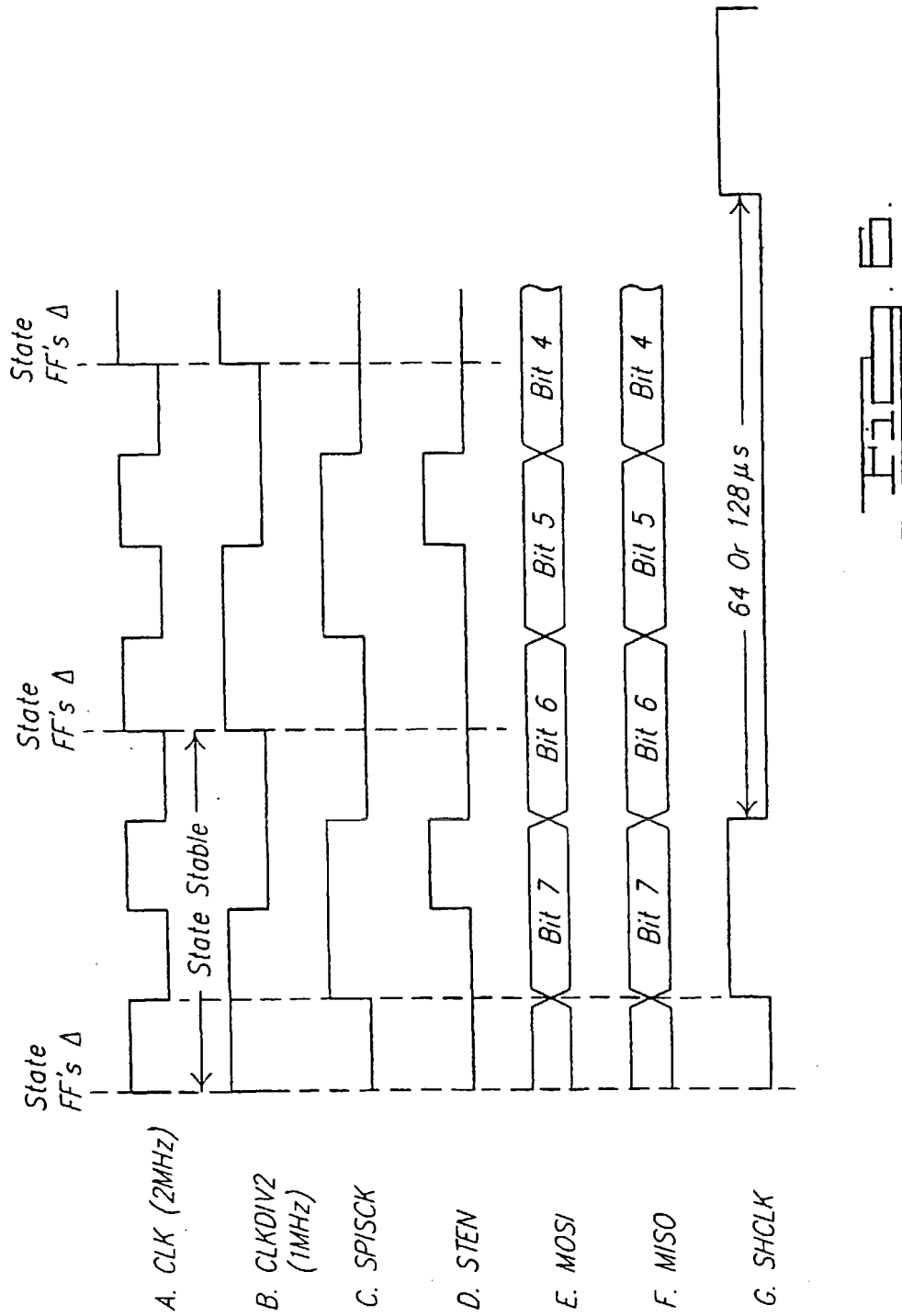
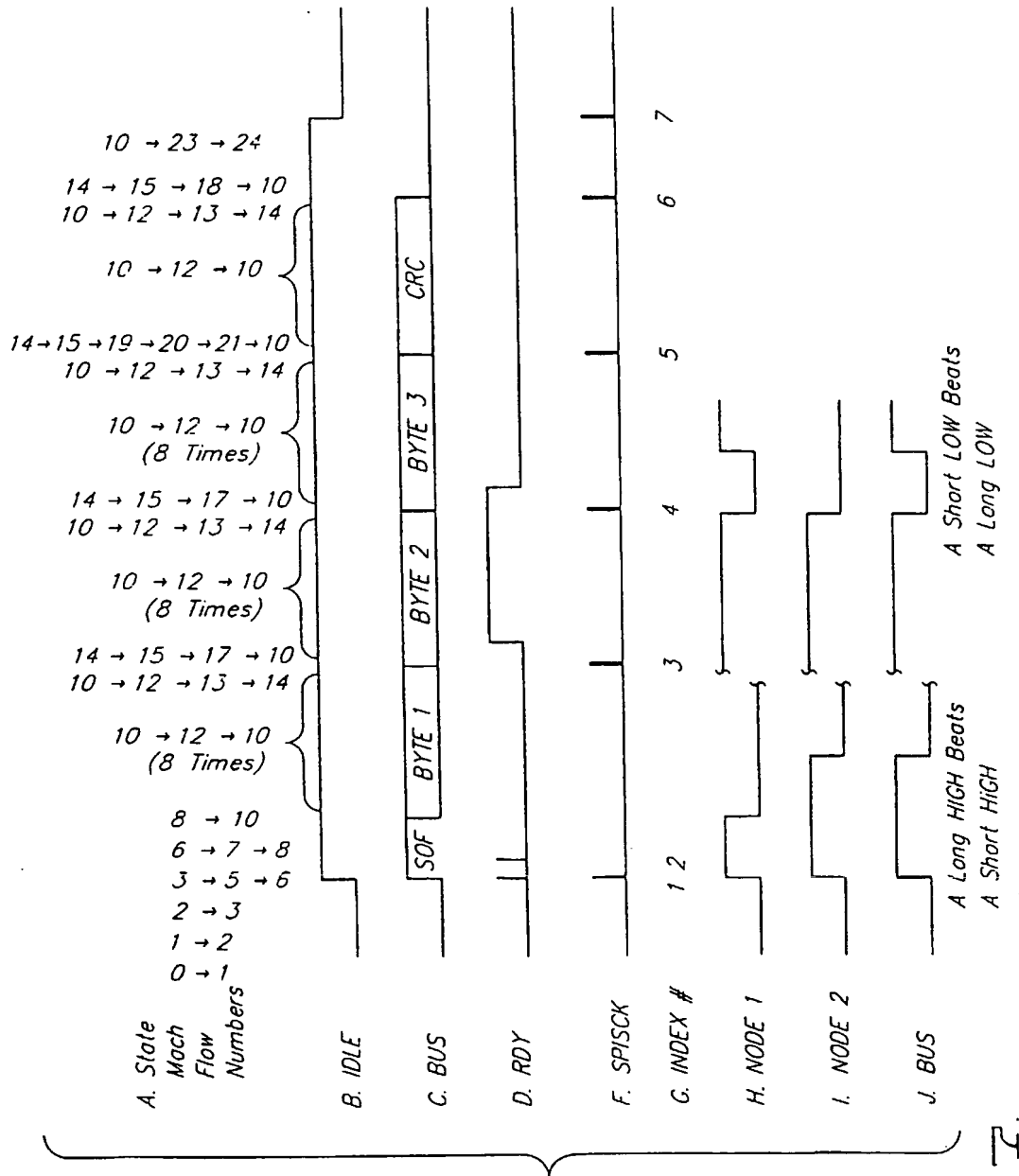
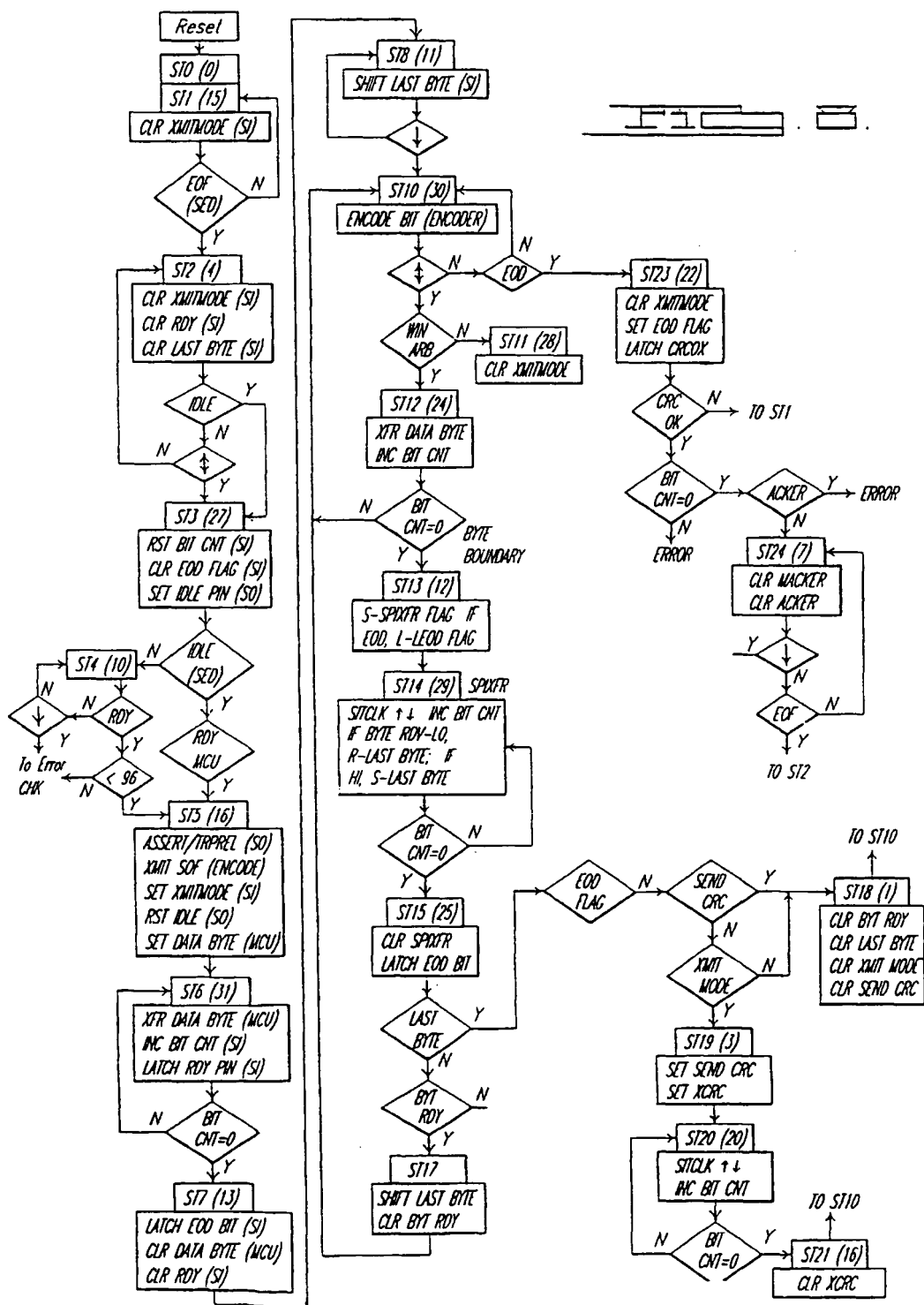
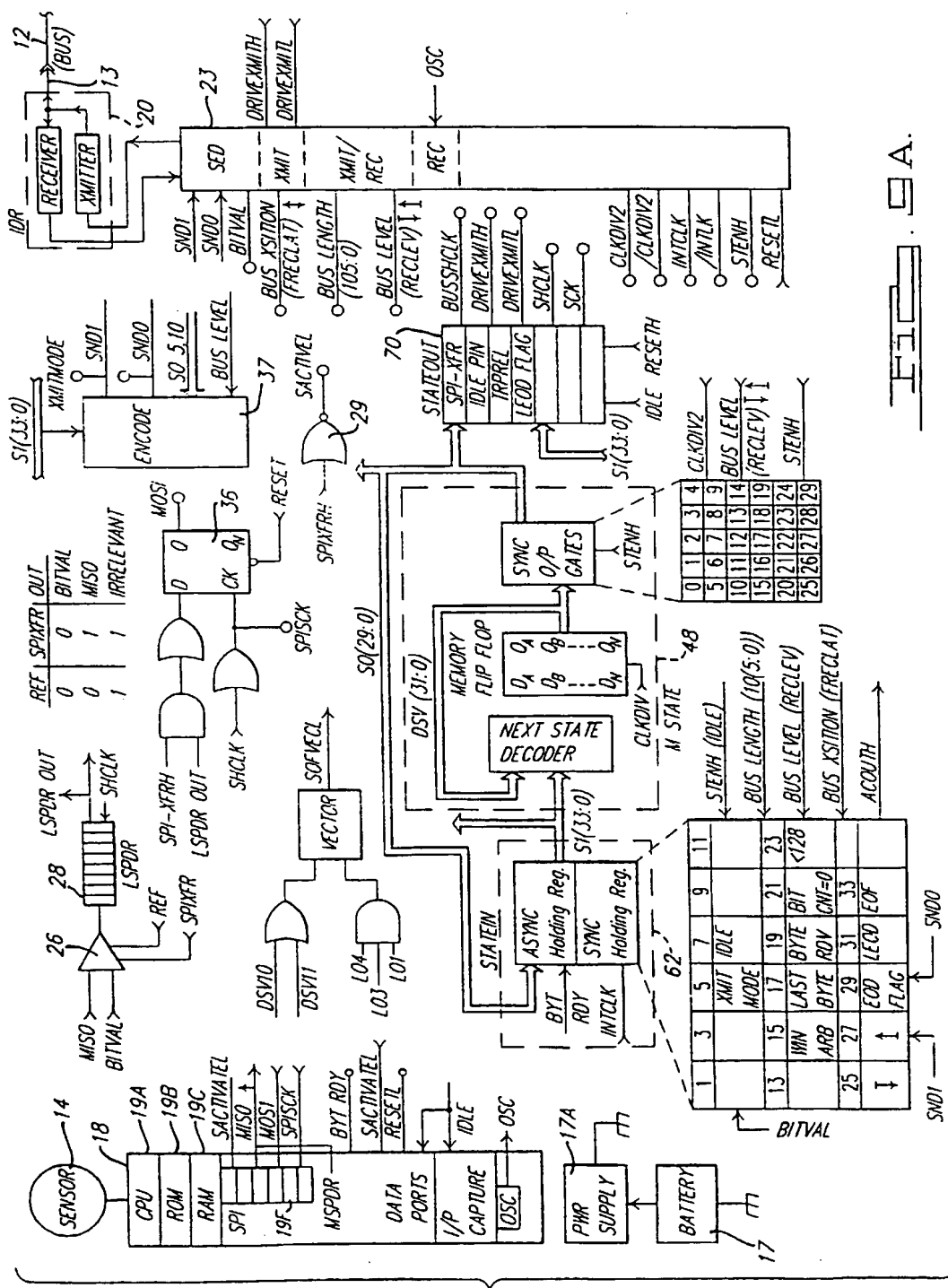
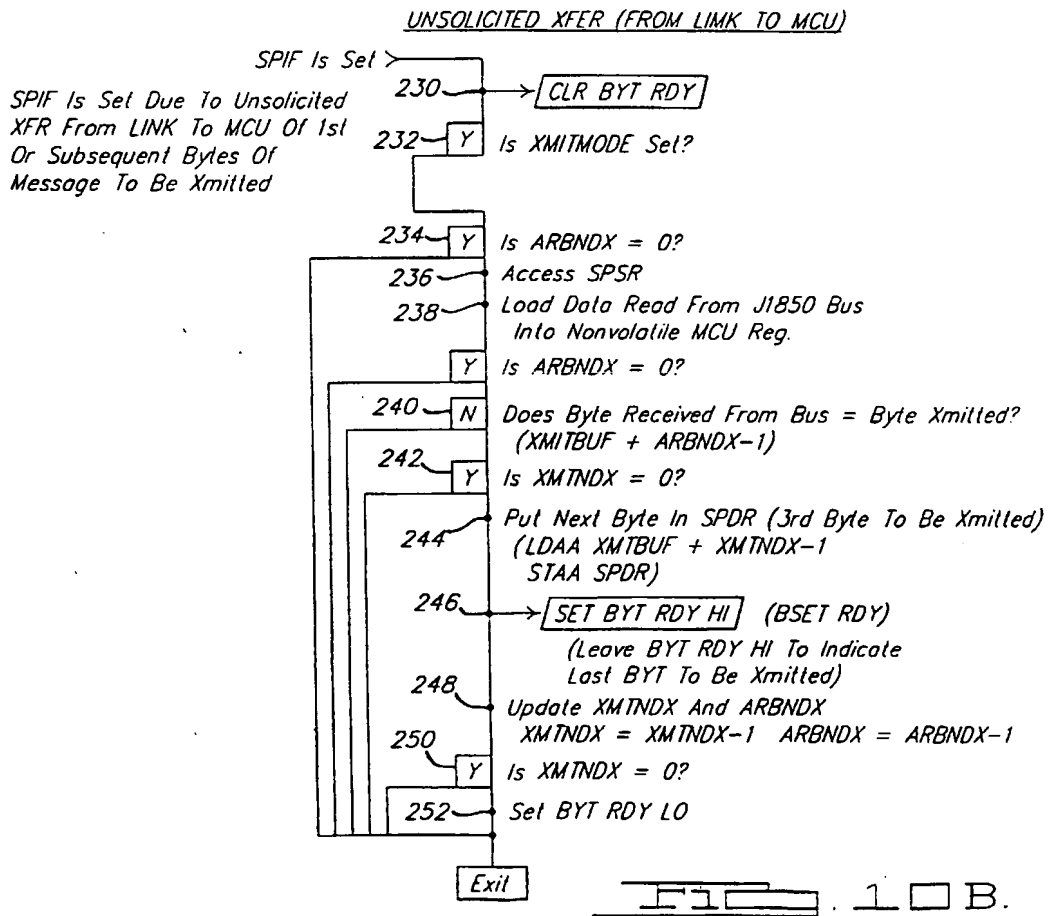
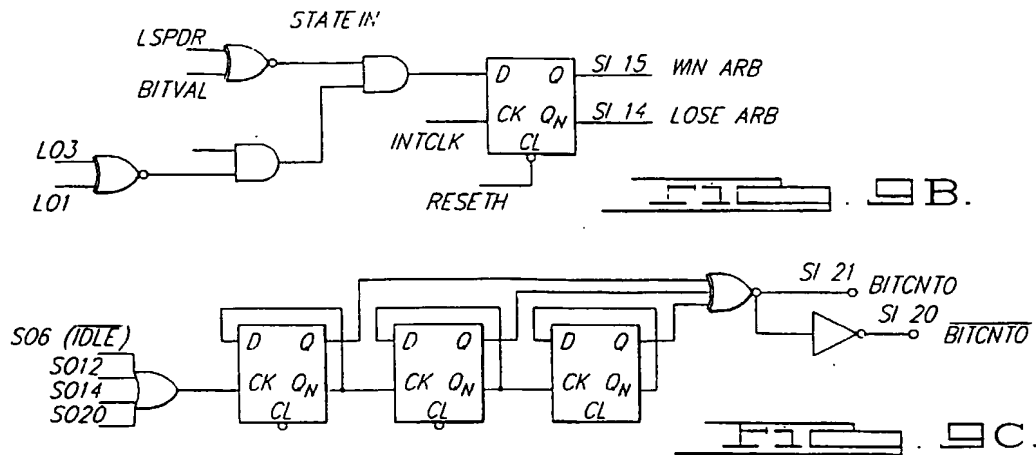


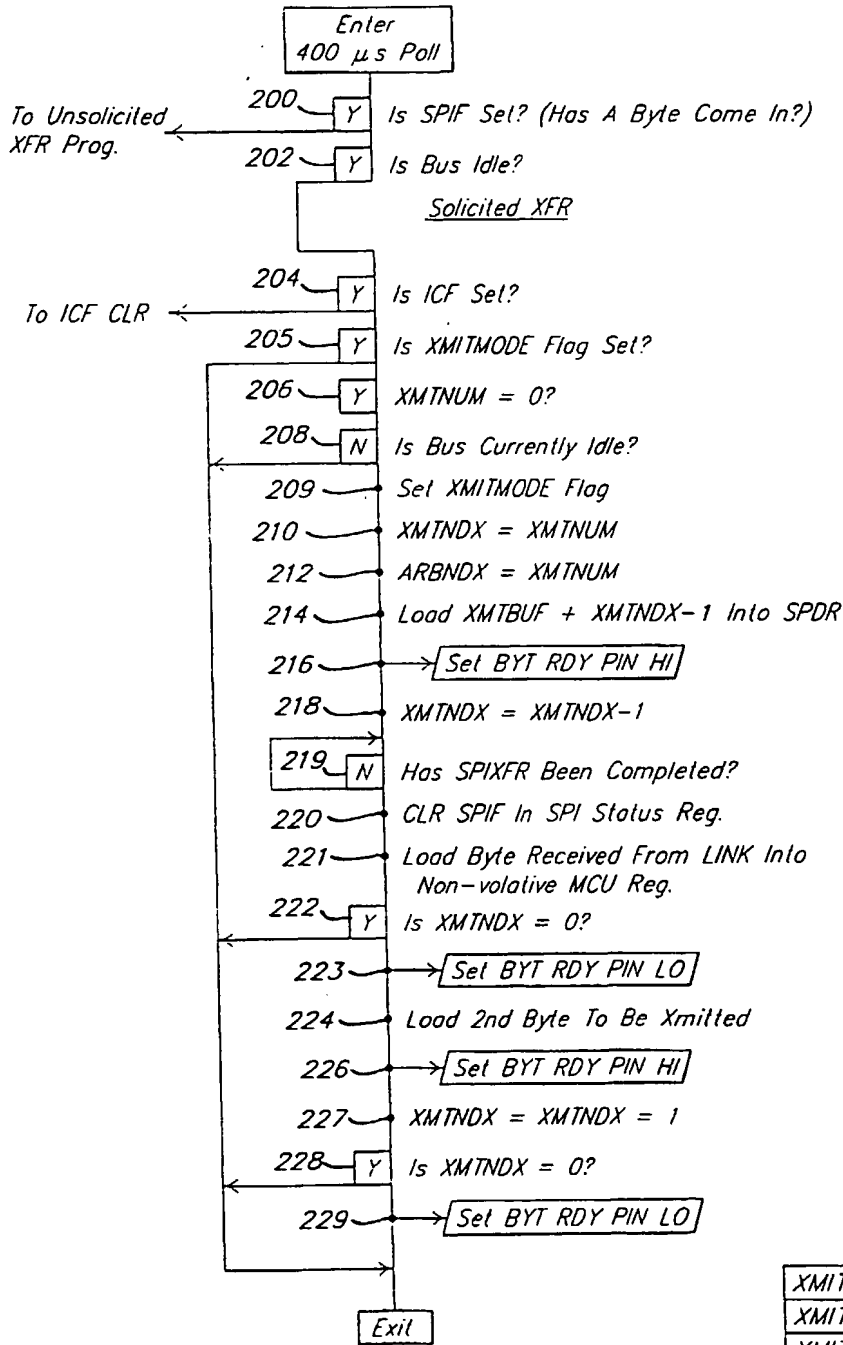
FIG. 5.











XMITBUF,0	DATA
XMITBUF,1	DATA
XMITBUF,2	ID

FIG. 1A.

FIG. 1C.